
Пример реализованной стратегии, на каждом ходу делающей ход в случайную клетку (C++):

```
#include <iostream>
#include <fstream>
#include <ctime>
#include <cstdlib>

using namespace std;

const int SZ = 20;
int field[SZ][SZ];
int numfree;
int freerow[SZ * SZ];
int freecol[SZ * SZ];

int main()
{
    srand((unsigned)time(NULL));
    int i, j;
    while (1)
    {
        for (i = 0; i < SZ; i++)
        {
            for (j = 0; j < SZ; j++)
            {
                scanf("%d", &field[i][j]);
            }
        }
        numfree = 0;
        for (i = 0; i < SZ; i++)
        {
            for (j = 0; j < SZ; j++)
            {
                if (field[i][j] == 0)
                {
                    freerow[numfree] = i;
                    freecol[numfree] = j;
                    numfree++;
                }
            }
        }
        if (numfree == 0) break;
        int pos = rand() % numfree;
        printf("%d %d\n\n", freerow[pos], freecol[pos]);
        fflush(stdout);
    }
    // чтобы программа не завершилась раньше времени
    scanf("%d", &i);
    return 0;
}
```

Пример реализованной стратегии, на каждом ходу делающей ход в случайную клетку (Pascal):

```
const SZ = 20;
var field : array[1..SZ, 1..SZ] of integer;
i, j : integer;
freerow, freecol : array[1..SZ * SZ] of integer;
numfree, pos : integer;
begin
randomize;
while true do begin
    for i := 1 to SZ do begin
        for j := 1 to SZ do begin
            read(field[i, j]);
        end;
    end;
    numfree := 0;
    for i := 1 to SZ do begin
        for j := 1 to SZ do begin
            if (field[i, j] = 0) then begin
                inc(numfree);
                freerow[numfree] := i;
                freecol[numfree] := j;
            end;
        end;
    end;
    if (numfree = 0) then break;
    pos := trunc(random() * 1000) mod numfree + 1;
    writeln(freerow[pos] - 1, ' ', freecol[pos] - 1);
    writeln;
    flush(output);
end;
read(i);
end.
```